Big Data Preventive Maintenance for Hard Disk Failure Detection

Chuan-Jun Su and Jorge A. Quan Yon

***Abstract*—In recent years, with the rapid advancement of cloud technology, data centers have always been considered one of cloud services most important aspect to evaluate, and its reliability and availability have been the focus of every IT engineer's attention. However, service interruption is the most important factor to consider for every data center, affecting the user experience, or causing loss in a business. Therefore, automated fault prevention and monitoring of data center will effectively improve the reliability of cloud services. Predictive maintenance differs from traditional maintenance process (i.e. routine maintenance and corrective maintenance), it evaluates the state by performing device condition monitoring, and according to the state, it predicts when maintenance should be performed.**

This research focuses on hard drive failure prediction, with big data analysis and machine learning technology, we have developed a Preventive Monitoring System (PMS). Utilizing Prognostics and Health Management (PHM) to identify the failure mechanism, and combining Self-Monitoring, Analysis and Reporting Technology (SMART) to identify early signs of abnormalities before the device fail. Finally, we use random forest algorithm to construct the predictive model. This research aims to develop a predictive monitoring system to provide device condition monitoring and fault diagnosis, thereby identifying the device malfunction and resolving it as soon as possible, keeping the system maintained at optimal condition.

***Index Terms*—Big data, preventive maintenance, failure prediction, hard drive, random forest.**

I. Introduction

As cloud computing and virtualization has emerged becoming new paradigm for computing at scale and provision of web services, the Information Technology (IT) infrastructure of service providers has become critical to the uptime of millions of users. The reliability of contemporary storage devices which have already become an integral part of the infrastructure plays an important role in the success of the infrastructure [1].

A reliable infrastructure is necessary in order to minimize any chance of disturbance. To ensure the functionality and interconnectivity, service providers must keep high standards of its hosted computer environment. Nowadays, developers and IT teams depend on secure, durable, and highly scalable cloud storage system for delivering continuous and pervasive web services. From a business perspective, as one of the most important components in storage systems - the hard drive, its failure might derive such financially devastating events as the loss of customer information, business transactions, and sales data, which could crumble the entire company.

Modern hard drives are in general reliable devices in which failure rate could below one percent according to many manufacturers' claim. However, studies have illustrated that failure rates can often be as high as 14 percent [2]. They are still vulnerable and have limited life span. There are frequently no warning signs for precaution before catastrophic failure of hard drives occurs. It is therefore critically important to develop a means of preventive maintenance for hard drives to provide such vital precaution.

A. Preventive Maintenance in Big Data

Reliability has always been an important aspect in the assessment of service and product. High reliability is usually accompanied by good product and process design. However, products will deteriorate over time regardless how robust the design is since they are providing service and operating in the real environment. Maintenance is an efficient way to retain or restore a piece of equipment, machine, or system to the specified operable condition to achieve its maximum life span.

Preventive Maintenance (PM) is a maintenance technique which is performed before failing. It conducts regular (or continuous) monitoring and failure prediction on the device components. Based on the possible causes obtained in the monitoring results, the PM pre-formulates the maintenance plan.

PM is a strategy for maintenance whereby breakdowns are avoided through activities that monitor equipment deterioration and undertake minor repairs to restore equipment to proper condition. Like human health screening, PM emphasizes that Prevention is better than cure. The PM approach is based on the maintenance condition and hence it needs accurate and complete information for optimal decision-making. As the big data technology is getting mature, systematic analysis of massive streaming data have been developed for exploring hidden patterns. With varying measures, PM has been developed and has an increasingly wide utilization in all engineering fields, such as vibration monitoring, temperature monitoring, and noise monitoring. In the field of computer management and intelligence, log file analysis is a common approach to monitoring the system status.

B. Cloud Prediction Tool

A Cloud prediction tool provides machine learning and pattern matching capabilities. With these capabilities, you can develop applications to perform tasks like categorizing an email as a spam or non-spam, assessing whether comments have positive or negative sentiment, or predicting the customer flow at a shopping mall on a given day. Fig. 1

Manuscript received October 23, 2017; revised December 25, 2017.

The authors are with the Department of Industrial Engineering and Management, Yuan Ze University, Taiwan (e-mail: iecjsu@saturn.yzu.edu.tw, s1038908@saturn.yzu.edu.tw).

doi: 10.18178/ijiet.2018.8.7.1085

shows a cloud prediction tool framework, it contains two stages: batch training and real-time predictions. At the beginning of batch training stage, in order to generate a machine learning model, we collect historical data as the data set, then build a prediction model based on it. Real-time predictions stage uses the data which comes from live stream. A revised model also trained when streaming date become historical data.

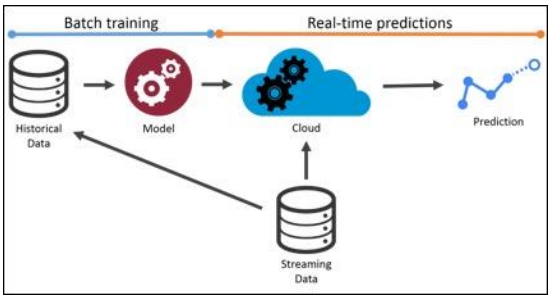

Fig. 1. Prediction tool framework.

C. Failure Prediction in Computer Science

Laprie definition of failures is commonly used [3]:

"A system failure occurs when the delivered service deviates from the specified service, where the service specification is an agreed description of the expected service."

The fact is that we are getting increasingly more dependent on computer systems, user expect and even demand that computer systems should never fail, still systems are inherently to failures. Failure results in high costs for users who may experience loss of service for some period of time or even loss of important and valuable data. One of the best ways to prevent a failure is to predict failures by analyzing the system logs and identifying the failure sign. In this research, we try to construct a predictive monitoring system with these research findings.

D. Research Objective

The objective of this research is to provide a tool that can assist the developers and IT teams when maintaining a large scale storage system, to warn users of impending drive failure. To solve the above problems, this research develops a framework of predictive monitoring for hard disk failure by analyzing the machine log files rather than using traditional statistical prediction method.

The development of this research involves in the following tasks:

1) The review of main factors in hard drive failure in literatures.
2) Retrieve and pre-process the data from the hard disk.
3) Train and verify the prediction model.
4) The construction of prediction system.
5) Predicting by using streaming data.

II. Literature Review

A. Big Data Analysis

Due to the constant generation of new information, today's world is witnessing the exponential growth of data. These data have to be collected, stored, and processed [4]. Big data analysis has the problem of analyzing, modeling and extracting information from huge amounts of data. Traditional methodologies only analyze historical data, until now, the requirement satisfied by rapid-growing data volumes, storage device cost reduction, technical software progress, and mature cloud environments, it has been evolved to capabilities as predicting features and creating the brand new business model.

B. Failure Precursor

A failure precursor is an event or series of events that is indicative of an impending failure [5]. A failure can be predicted by correlating changes in monitored precursor parameters. Failure precursor parameters can be identified by the following steps: performing the FMMEA, accelerated aging, physical degradation analysis, and electrical characterization.

By identifying the precursor parameters and monitoring them, the extent of deviation or degradation from expected normal operating conditions can be assessed. This information can be used to provide advanced warning of failures and for improving qualification of products.

Reference [6] presented a PHM approach for power supplies. First, they analyze historical data to identify precursor parameter, then conduct experiments under different environmental and usage conditions to establishment baseline. Finally, precursor parameter identification for one switch-mode power supply (SMPS) was carried out. The approach process is represented in Fig. 2.

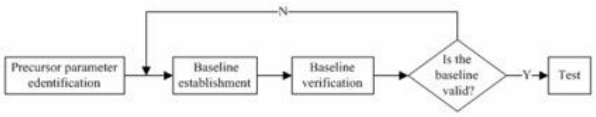

Fig. 2. An approach to power supply prognostics.

C. Preventive Maintenance in Big Data

Preventive Maintenance (PM) maintain devices in satisfactory operating condition by monitoring for avoiding devastating events that occurred before they reach the end of their useful life. Big Data provide data continuously come from multiple sources in real-time, this enabled PM to fast diagnosis and repair large number of devices, reduce mean time to repair (MTTR), and keep equipment maintenance cost down. Furthermore, collecting data incessantly and analyzing statistics for continuous improvement of device performance. Examples of applications are as following:

Reference [7] proposed a fraud management and a fraud prevention architecture based on implementing fraud policies, which can process real-time data streams, attempt to detect fraud proactively, block suspicious transactions.

Reference [8] studied the problem of matching both static and dynamic patterns over stream time-series data in stock data monitoring and propose a novel multistep filtering mechanism.

Reference [9] proposed a novel measure for intrusion detection and surveillance by using pure time series data stream.

Reference [10] utilized the Bayesian robust principal component analysis (RPCA) approach and developed a new method for detecting traffic events that impact road traffic conditions. This method processes data streams in an

incremental way with small computational cost, and it is suitable to detect events in an online and real-time manner.

D. Self-monitoring and Reporting Technology (SMART)

Every hard disk drives (HDDs) and solid-state drives (SSDs) include Self-Monitoring, Analysis, and Reporting Technology (SMART), which detects and reports internal information about the drive. The first time hard disk monitoring technology was used is by IBM in 1992. It was named Predictive Failure Analysis (PFA) technology. The technology measures several key device health parameters of the drive unit (e.g., the flying height of heads) and evaluating them within the drive firmware.

The existing monitoring methodology is by thresholds analysis, which are predefined by the manufacturer of the drives. The device firmware compares the thresholds with the measured parameters, if a SMART attribute drops below its threshold, the drive may not able to meet its specifications and fail in the nearly future. Note that attributes change slowly over time and are helpful attempts to diagnose the life span of a hard drive. Some studies have tried to improve the thresholds methodology, and tried to figure out the critical factor of the drive failure.

Reference [11] discussed two Bayesian methods to predict disk drive failures based on measurements of drive internal conditions. They performed sensitivity and specificity analysis to Quantum SMART dataset. The result shows that drive failure has positive correlation with power-on hours, read error rate and spin-up time attributes.

Reference [12] proposed improved methods for disk-drive failure prediction. By observing SMART warning of hard drive failure, intervals of parameter value are analyzed in order to enhance the accuracy of disk-drive failure detection rate.

Reference [13] presented a case study of hard drive failure predicting by using SMART attributes. After comparing the performance of support vector machines (SVMs), unsupervised clustering, and non-parametric statistical tests (rank-sum and reverse arrangements), the rank-sum method have better performance than other methods.

Reference [14] presented data collected from detailed observations of a large disk drive population in a production Internet services deployment. They analyzed the data, which was collect from more than one hundred thousand disk drives, which are a combination of serial and parallel ATA consumer-grade hard disk drives, ranging in speed from 5400 to 7200 rpm, and in size from 80 to 400 GB, and tries to find the reasons cause hard drive failure. The result shows that several parameters correlate highly with failure, and are suitable for being disk drive health indicators.

Reference [15] constructed a decision support system which can detect failures and analyze cause. Discovering rules from disk events and performing black-box model to improve the accuracy of disk-drive failure prediction.

E. Machine Learning

One of the central feature of intelligence is the ability to learn, which makes it an important concern for artificial intelligence (AI). Machine learning is a subject within computer science [16] that examines how to write programs that can learn. Arthur Samuel in 1959 defined machine learning as:

"field of study that gives computer the ability to learn without being explicitly programmed." [17]

Another definition of machine learning comes from [18]. This book defines a well-posed learning problem as:

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

Applications that typically use machine learning techniques, such as data mining, email filtering, computer vision, natural language processing(NLP), optical character recognition(OCR), biometrics, search engine, medical diagnosis, credit card fraud detection, and speech recognition.

In data mining, machine learning is used for prediction or classification. With machine learning, the computer makes a prediction and then, based on feedback whether it is correct, "learns" from this feedback. It learns through examples, domain knowledge, and feedback. When a similar situation arises in the future, this feedback is used to make the same prediction or to make a completely different prediction. Statistics are very important in machine learning programs because the results of the predictions must be statistically significant and must be better than a naive prediction. Sample data is used to train the machine learning system to properly perform the desired task. A model is generated after the training section. When new data is input, it can perform prediction using the model.

There are two types of machine learning: supervised learning and unsupervised learning. Supervised learning is given by a training set of data with correct answers, the computational model successively applies each entry in the training set. In other words, a supervised approach learns by example. Based on its capability to correctly handle each of these entries, the model is changed to ensure that it works better with this entry if it were applied again. Given enough input values, the model will learn the correct behavior for any potential entry. With unsupervised data, data exist but there is no knowledge of the correct answer of applying the model to the data.

TABLE I: INSTANCES WITH KNOWN LABELS

Data in standard format					
case	Feature 1	Feature 2	…	Feature N	Class
1	xxx	x		xx	good
2	xxx	x		xx	good
3	xxx	x		xx	bad
…					…

Supervised learning using the same set of features in every instance with known labels in any dataset. The features can be continuous, categorical or binary (see Table I). On the other hand, unsupervised learning using instance with unknown labels [19].

This research aims to build a prediction system which is trained on past behavior to predict the disk failure. One of the components of the prediction system is a classifier. Ensemble learning has several classifiers, such as Neural Networks, Support Vector Machine (SVM) and Random Forest. Classification algorithms must be able to handle big data problems; by comparison, Random Forest's performance compares very favorably with that of other classification

algorithms. Benefits of random forests are as following [20]:

1) It is unexcelled in accuracy among current algorithms.
2) It runs efficiently on large data bases.
3) It can handle thousands of input variables without variable deletion.
4) It gives estimates of what variables are important in the classification.
5) It generates an internal unbiased estimate of the generalization error as the forest building progresses.
6) It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

F. Random Forest

In this research, random forest was selected based on its availability for preventive monitoring hard drive failure. Random forest algorithm was presented by [21] which includes classification and regression tools [22], [23]. It constructs a number of decision trees at training time by using bootstrap sampling [24], tree predictors are sampled independently with the same distribution for all trees in the forest. Random forest outputting a class as the result which is the mode of the classes output by individual trees.

Bootstrap aggregating (also called bagging) is a model averaging approach that is used to improve the stability and accuracy of machine learning algorithms. Fig. 3 illustrates the basic principles of bagging. The data was modeled to the blue circles, we assume there are noisy data in the dataset. Each individual learner is shown as a gray line curve by partial sampling (a weak leaner). The red line can be seen to be a much better approximation to the underlying data (the ensemble strong learner).

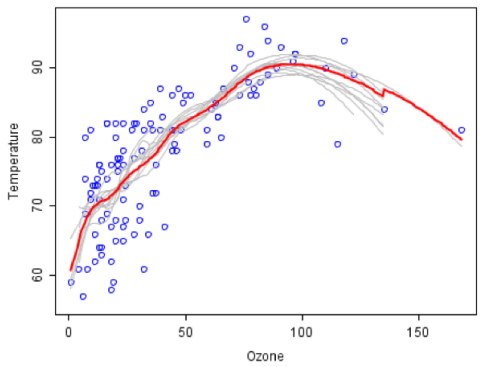

Fig. 3. The basic principles of bagging.

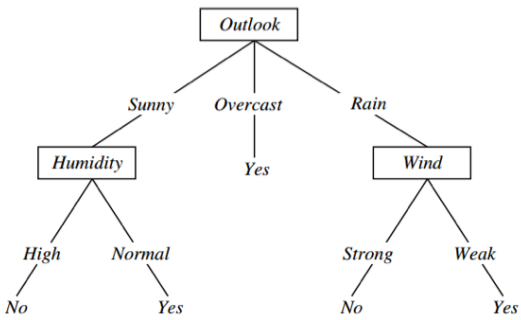

Fig. 4. A decision tree example.

Random Forest starts with constricting decision trees, which uses a tree-like structure or model to support decision making in machine learning [25]. The arcs emanating from each node represent each possible answer to the associated question. Each leaf node represents a prediction of a solution to the problem under consideration [26]. Fig. 4 below shows an example was provided by [18].

In this example, a decision tree was constructed to solve the play ball problem. The tree gives advice based on weather conditions. For example, if the outlook is rain and the wind is strong, then it's probably not OK to play.

The construction of RF is described in the following main steps:

a) For

$$b = 1 \; to \; B: \tag{1}$$

(a) Draw a bootstrap sample Z^* of size N from the training data.

(b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

1) Select m variables at random from the p variables.
2) Pick the best variables/split-point among the m.
3) Split the node into two daughter nodes.

b) Output the ensemble of tree $\{T_b\}_1^B$.To make a prediction at a new point x:

Regression:

$$\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b\,(x). \tag{2}$$

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random forest tree. Then

$$\hat{C}_{\mathrm{rf}}^B(x) = majority\ vote\{\hat{C}_b(x)\}_1^B. \tag{3}$$

The way Random Forest works is that it does random selection on two levels:

1) A subset of the data
2) A subset of features to split that data

Both these subsets can overlap.

Random forest utilizing bootstrap aggregating to sub-sample data sampled used for training. On average, there are about one-third of examples that are not in the bootstrap set. They are called Out-of-bag (OOB) samples [21]. The OOB error is the mean prediction error on each training sample and it estimates the generalization error of random forest classifier.

Random forest has been widely used in various applications because its superior performance and its relative simplicity in design.

Reference [27] presented an aircraft engine fault diagnosis system. They investigated design details of random forest classifier and evaluates its performance. They also make some efforts on investigating strategies for improving random forest performance specifically for aircraft engine fault diagnosis problem.

Reference [28] presented a modeling framework that combined random forest algorithm and geographic information systems for identifying suitable sites and quantifying site-specific yields for the exploitation of an aquaculture species.

Reference [29] developed a random forest based methodology to assess the current health and predict remaining useful life of both trucks and wheels of a railcar by

fusing measurements from the three types of wayside detectors. They further showed that, MissForest, a random forest based nonparametric imputation method, which can effectively handle missing data in wayside detector readings.

Reference [30] tested a random forest model for predicting soil USCS classifications from soil property variables. The result showed that important variables for predicting USCS code from available soil properties were soil textures, percent organic material, and available water storage.

III. Research Methodology

With the prevalence of computers, new technology on storage devices such as flash memory-based solid-state drives (SSDs), which have higher data transfer rates and better reliability, have been developed. However, hard disk drives (HDD) remain the dominant medium for storage in organizations because of their cost advantages. The need for generating precautions before the failure of mass-produced HDD to avoid devastating events in enterprises and personal computing environment is greater than ever. The solution to this need may lie in an innovative and rapid-growing technology known as Predictive Analytics of Big Data.

In this section, we present a predictive analytics framework (PAF-HD) based on Random Forest for preventive monitoring of HDD failure.

A. The Overview of PAF-HD

The proposed Predictive Analytics Framework (PAF-HD) architecture comprises three modules as illustrated in Fig. 5: 1) the Extract, Transform and Load (ETL) module, 2) the Machine Learning (ML) module, and 3) Prognostics and Health Management (PHM) module. Each module encapsulates two components, the details of modules resided in the PAF-HD are described in the following:

The ETL module contains Failure Mode (FM) and Extraction and Storage (ES), ML is formed by Feature Engineering (FE) and Generate Model (GM), whereas PHM comprises Condition Monitoring (CM) and Warning System (WS).

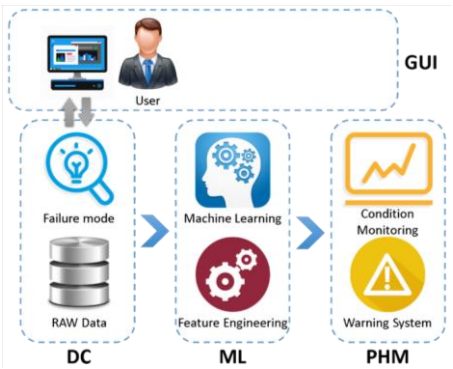

Fig. 5. PAF-HD architecture.

Fig. 6 illustrates the workflow of PAF-HD driven by the interactions of the three modules.

B. Backblaze SMART Dataset

The data used in this research come from real-world collection by Backblaze, Inc. (https://www.backblaze.com/). Backblaze provides online encrypted backup services for individuals, families, organizations and businesses in over 140 countries. Backblaze cloud backup has backed up over 200 million GB of data for Mac and PC laptops and desktops. In order to provide this service, Backblaze deployed over 56,224 hard drives ranging from 1.5 terabyte to terabyte in size at the end of 2015. As a highly open company, Backblaze publishes detailed information on the Storage Pod hardware, the Vault software, and the reliability statistics behind the hard drives it uses.

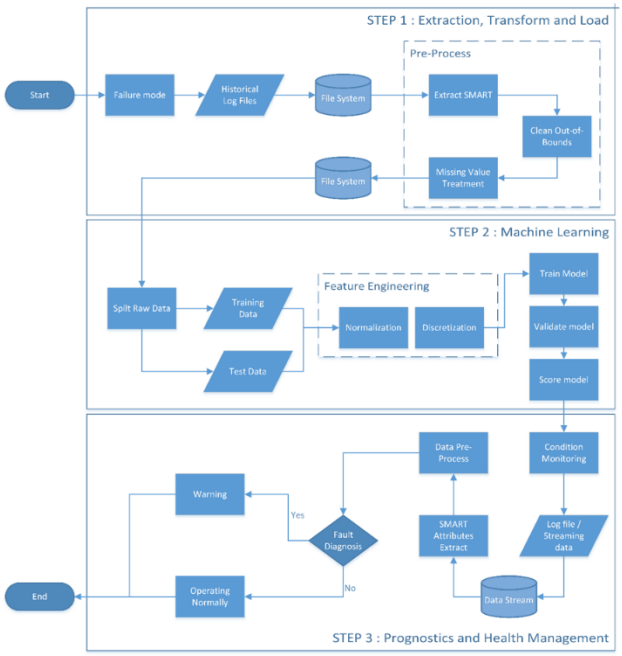

Fig. 6. PAF-HD workflow.

Each day, Backblaze data center takes a snapshot of operational hard drive. The snapshot includes basic drive information along with the SMART statistics reported by that drive. A record of one drive is saved as one row of data, and all of the drive snapshots are collected into a CSV format file for a given day. The CSV file contains the columns, which are shown as following:

1) Date – The date of the file in yyyy-mm-dd format.
2) Serial Number – The manufacturer-assigned serial number of the drive.
3) Model – The manufacturer-assigned model number of the drive.
4) Capacity – The drive capacity in bytes.
5) Failure – Contains a "0" if the drive is OK. Contains a "1" if this is the last day the drive was operational before failing.
6) SMART Stats – 90 columns of data, that are the Raw and Normalized values for 45 different SMART stats as reported by the given drive. Each value is the number reported by the drive.

C. The Module 1: ETL

The ETL module encompasses two fundamental components: Failure Mode (FM) and Extraction and Storage (ES). Backblaze defined a drive failed as the following:

1) Totally stop working.
2) It has shown evidence of failing soon.

When a drive has one of above condition, it is removed from a Storage Pod and replaced. To this, Backblaze marks the drive as failed. A totally stop working drive appears when the drive is physically dead (e.g. will not power up), it shows

no response to console commands or the system cannot read or write the drive.

For determining if a drive is going to fail, Backblaze use SMART statistics as evidence. Removing a drive before catastrophic failure occurs or impedes the operation systems.

Random forest has many advantages, one of it is handling great amount of input variables without variable deletion. We take advantage of it, when dealing with the high-dimensional data, we do not need to select variables.

The Extraction and Storage (ES) component is responsible for gathering SMART data from system log files, pre-processing includes SMART extraction, out-of-bounds cleaning and missing values treatment. The module performs as below:

1) Data extraction — extracts data from data sources.
2) Data transformation — structure the data for the purposes of analysis.
3) Data loading — loads it into the final target.

The raw data consist of many information that not all of them are needed when generating the machine learning model. In order to extract these selected columns, an extraction process is required. Besides, in a few cases the drive reported a weird value. For example, the RAW value of SMART 9 (Drive life in hours) reported a value that would make a drive 10+ years old, which was not possible. In other words, these special attributes need to be processed before analyzing it. In this research, we use a chain of filters to deal with the data, detailed in fig. 7:

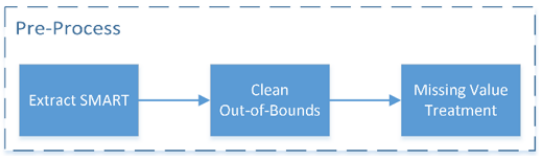

Fig. 7. Pre-processing structure.

1) SMART Extraction: extract SMART attributes from the raw data.
2) Clean Out-of-Bounds: perform a bound check before using the data.
3) Complement Missing Value: sometimes, those values are missing due to the device problem. Complement Missing Value deals with missing data by backward filling the values.

In order to implement the PAF-HD we take advantage of the benefits offered by the programming language Python to process various tasks like loading the dataset, learning and predicting, model preserving, and model validation.

The PAF-HD database is built upon Hadoop Distributed File System (HDFS), a distributed file system designed for distributed storage and distributed processing. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. The PAF-HD database keeps a complete time-ordered history of the log files in each hard drive's life.

D. The Module 2: ML

The proposed system for the second module basically is composed of two main components, Feature Engineering (FE) and Generate Model (GM). Once we have defined the hard drive failure mode and prepared the SMART data, we need to apply machine learning algorithms to the data in order to solve the problem - predictive monitoring for hard disk failure. Fig. 8 shows the overview of fitting a machine learning model.

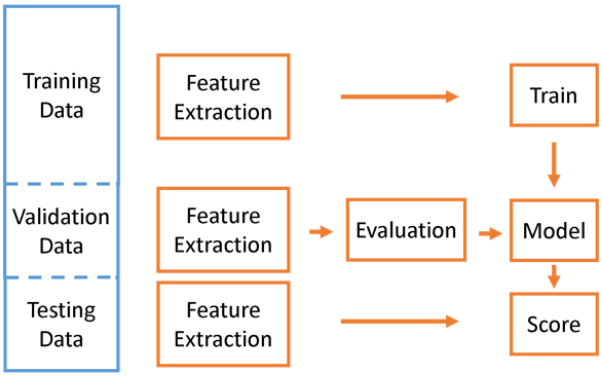

Fig. 8. Fitting a machine-learning model.

The Feature Engineering (FE) component converts raw data into training data and test data, then formulates and stores the features for the machine learning generation. The features are referenced to the failure mechanisms analysis. Two defined of feature engineering are as below:

"Coming up with features is difficult, time-consuming, requires expert knowledge. Applied machine learning" is basically feature engineering." [31]

" When working on a machine learning problem, feature engineering is manually designing what the input x's should be." [32]

In this research, we use a chain of filters to deal with the data, detailed in Fig. 9:

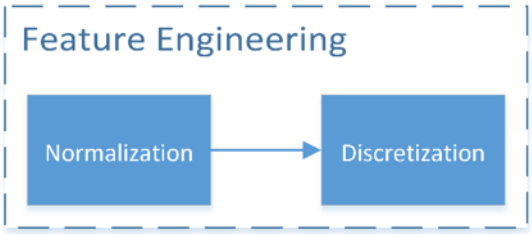

Fig. 9. Feature engineering structure.

1) Normalization: adjusting values measured on different scales to a notionally common scale.
2) Discretization: transferring continuous value to discrete value.

E. Generate Model (GM)

In this step, the SMART attributes in the file system are ready to train the model. Scikit-learn provides a set of algorithms for fitting machine-learning models to a large data sets and perform related statistical processing. Using the capability of scikit-learn package for prediction is a three steps process: (1) initializing the model, (2) fitting it to the training data, and (3) predicting new values.

The output is an array, which includes a length equal to the number of hard drives devices in the test set and a prediction of whether they have failed.

F. The Module 3: PHM

For our last module in the implementation of the PAF-HD framework, we propose a more fine-grained method to monitoring devices. In order to gather and process the data, an infrastructure is required. The System Health

Infrastructure, a large distributed software system, was been described by [33], which collects and stores hundreds of attribute-value pairs from all of Google's servers. It also provides the interface for arbitrary analysis jobs to process that data. We use the architecture of the System Health Infrastructure to illustrates the infrastructure of PAF-HD. The infrastructure concept is shown in Fig. 10.

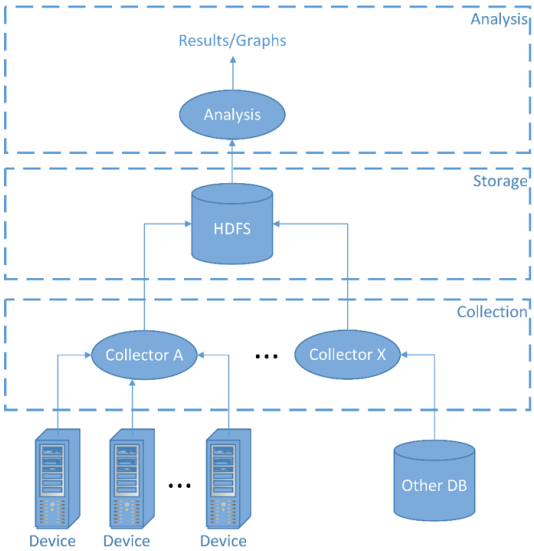

Fig. 10. PAF-HD infrastructure concept.

The PAF-HD infrastructure contains a data collection layer, which collects data via historical data or streaming data from each device, A distributed file system and an analysis server.

Until now we have trained our predictive model, to further the original monitoring system, we propose a two-step approach:

1) The Value Prediction (VP) component, a background service which has been set up at the device, sending stream data to the analysis cloud.
2) The Warning System (WS) component running at the analysis server, which receive the stream data from the device, analyzes and reports the device condition to the user.

G. *Value Prediction (VP)*

In order to collect the SMART data from the hard drive, a hard drive disk utility software is needed. In this research, we use smartmontools as the monitoring software (available at https://www.smartmontools.org/). Smartmontools consist of two utility programs (smartctl and smartd), which is a tool for monitoring SMART data on modern hard disk and solid-state drives. It allows you to inspect the drive's SMART data to determine its health, as well as run various tests on it.

The VP module use Smartmontools monitoring the device and sending stream data as a daemon, which runs on every device and gathers SMART data, the usage of the daemon is very light, so it will not interfere with the device operating.

With the application functions, hard disk drive status was also retrieved into a log file. Log file contains the device information, SMART attribute values and self-assessment test result. All log files are stored in the file system for feature inquire and use.

In order to minimize the usage of the device, the implementation of analysis process is built at analysis server. Once smartmontools logging the device status, the CM component send data as a stream.

The Warning System (WS) represents streaming data from device source. The WS process is shown at Fig. 11.

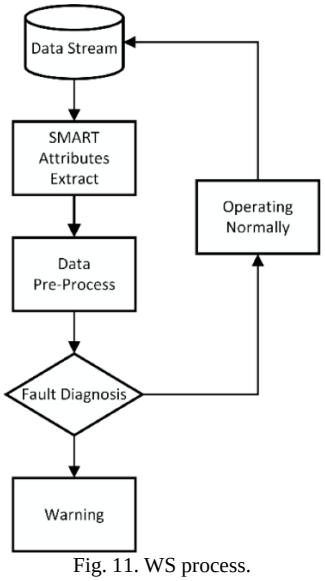

Fig. 11. WS process.

Device status receive from CM is necessary to be processed. The steps are as follows:

1) SMART attributes extracted from data stream.
2) Data pre-processing, includes Feature Selection, Clean Out-of-Bounds, Complement Missing Value, and Normalization.
3) Do the fault diagnosis. Make failure prediction based on the model generated by GM module. If result shows "Yes", then, a warning message shows up.

IV. Implementation

To demonstrate the feasibility of the proposed approach a prototype is developed, which embodies the generalized and functional design concepts for the PAF-HD system. The scenarios that cover the two phases of the prediction cycle are used to demonstrate the PAF-HD's comprehensive capability in accomplishing the research objectives illustrated in the chapter1.

A. *Hardware and Software Configuration*

The PAF-HD is implemented on the Linux platform. In order to derive the failure prediction of drive disks, a cloud analytics engine that needs to be built. The cloud environment in PAF-HD is built on Apache ecosystem components, including Hadoop (http://hadoop.apache.org/) distribution and Spark (http://spark.apache.org/), which helps us process real-time data by taking advantage of cost-effective storage, and lightning-fast data processing. The two big data frameworks are complementary, and work together very effectively as a big data system.

Apache Hadoop requires Java, we need to install the Java Runtime Environment (JRE). However, we recommended installing the Java Development Kit (JDK) because certain tools have a dependency on JDK for compiling generated code at runtime, and will not function without a JDK.

Spark jobs will have to read input data from an external storage system, a distributed file system designed for distributed storage and distributed processing called Hadoop

Distributed File System (HDFS) was used.

Apache Spark currently supports multiple programming languages, including Java, Scala, Python and R language. In this research, applications were coded in Python. By using Spark Python API (PySpark), we are able to interact with Spark through Python.

Smartmontools (https://www.smartmontools.org/) contains two utility programs (smartctl and smartd) to control and monitor storage systems using the SMART built into most modern disks. These utilities will provide advanced warning of disk degradation and failure. Smartmontools can run on any operating system such as Linux, Mac OS X, Microsoft Windows, etc. We use Smartmontools to monitor the health of a system's disks. The overall hardware and software configurations of PAF-HD are summarized in Table II.

TABLE II: PAF-HD'S HARDWARE AND SOFTWARE CONFIGURATIONS

Hardware Environment Configuration	
CPU	Intel Xeon CPU E5-2403 v2s
Memory	8G
Operation System	Ubuntu 14.04
Runtime and Implementation Configuration	
Python Version	Python 2.7.11
Drive monitor tool	smartmontools 6.2
Java Version	Java 1.7.0_95
Hadoop Version	Hadoop 2.6.0
Scala Version	Scala 2.10.4
Spark Version	Spark 1.4.0

B. Initiate PAF-HD System

Before we can run the system platform for the first time, we have to deploy a Spark server. Spark can run over several existing cluster managers currently: Standalone, Apache Mesos and Hadoop YARN. Compared to other Spark deploy mode, Spark on YARN is the most promising one.

In this research, we deploy Hadoop runs on a single-node in a pseudo-distributed mode and submit Spark applications to the YARN ResourceManager in client mode. Spark applications runs as independent sets of process on a cluster, and request resources from YARN.

C. Scenario 1 — Batch Training

BackBlaze provide personal and business online backup service. A summer intern, Alex, has been working hard with maintenance team in a data center, making sure that all the drives are spinning. Each day, they take a snapshot of each operational hard drive. This snapshot includes basic drive information along with the SMART statistics reported by that drive. The data center maintenance team already have some insight into detecting hard drive failure method, they know which drive has shown evidence of failing soon. However, traditional statistical has accuracy and timeliness problem. Alex looks for more innovation method, and come upon the PAF-HD system. So they decide to try the PAF-HD system to gather the opinionated resource to monitor hard disks and predict failures.

He follows the instructions for the initial setup, deploys Hadoop with Apache Spark, uploads the raw data to the HDFS. After finishing upload, he opens a web browser and access http://namenode:50070/. He checks the raw data integrity to ensure that there are no data missing problems occur during uploading.

For model building, Alex use the last two years of hard drives data. According to the Table 4, he decides to focus on model ST400DM000 which has the most failures in Q1 2016. First, he filters out specific records with the same model name. Consider the hidden pattern in the timeline, Alex manually adjust some records. For those failure columns record with 1 (drive is failing), he picks up past three month records, marks as 1, the remaining part are record with 0 (drive is OK). The transform concept is illustrated in Fig. 12.

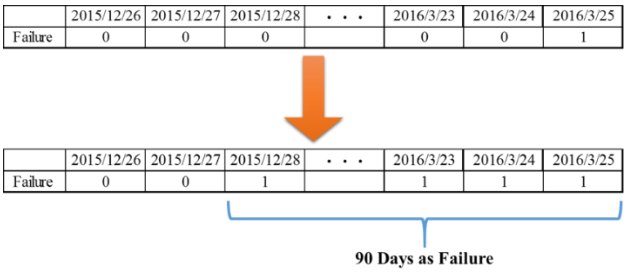

Fig. 12. The transform concept.

Now Alex has total 64539 records, failure records are 16862 and normal records are 47677. In order to balance the proportion, he takes random samples on the normal records. Then he has both 16862 failure records and normal records.

The PAF-HD provides data ETL tools, all those tasks can be completed easily.

The next step is model building. The PAF-HD system use machine learning algorithm to make predictions, currently algorithm offers Random Forest. The model which trained by the script above is saved on the HDFS so that it does not need to do model building every time and can load out the model for feature use.

The following two parameters are the most important for the RF model, and turning them can often improve performance:

1) n_estimators: The number of trees in the forest.
2) max_features: The number of features when fit is performed.

After several attempts, Alex gets some result as the following:

Fig. 13. Error rates.

Fig. 13 shows the error rate changes when trees are increasing. It shows that the model has low error rate with 200 trees. So Alex set the n_ estimators as 200.

For the other parameter, max_features, which means the number of feature considered at each internal node, Alex follows the instruction recommend and sets as 'sqrt'. The program will set the parameter as the square root of total features. The results are shown in Fig. 14.

```
        OOB estimate of  error rate: 13.29%
Confusion matrix:
     0   1 class.error
0 182   7  0.03703704
1  37 105  0.26056338
```

Fig. 14. Performance evaluation.

When the model is applied to new observations, the answers will be in error 13.29% of the time. That is, the accuracy is 86.71, the performance is OK.

D. Scenario 2 — Real-time Predicting

Stella, another summer intern in BackBlaze data center maintenance team. She is working with a batch of new hard drives which are newly arrive. Her job is to get drives online and connect them to the PAF-HD system that they have tried recently.

First, she installs the PAF-HD client on the manage operating system. The small customer module includes two software: Python and Smartmontools.

```
serverIP = '140.138.142.207'

from multiprocessing.connection import Client
c = Client((serverIP, 25000), authkey=b'peekaboo')

c.send(SMART_stats)
print 'Got:', c.recv()
```

Fig. 15. The partial code of the client script.

She opens the client python script of PAF-HD as shown in Fig. 15, type in analysis server IP and authorization key from the system administrator. When Stella runs the program, it creates a connection on the bound socket to the server.

The server script runs on server using a socket for listening connections as shown in Fig. 16. If socket gets connections and receive, it returns "success" message, otherwise it returns "Connection closed".

```
from multiprocessing.connection import Listener
import traceback

def echo_client(conn):
    try:
        while True:
            msg = conn.recv()
            print msg
            conn.send('success')
    except EOFError:
        print('Connection closed')

def echo_server(address, authkey):
    serv = Listener(address, authkey=authkey)
    while True:
        try:
            client = serv.accept()
            echo_client(client)
        except Exception:
            traceback.print_exc()

echo_server(('', 25000), authkey=b'peekaboo')  #25000
```

Fig. 16. The partial code of the server script.

```
connection accepted from ('140.138.142.18', 63660)
Predict Done
result = [0]
############################
Catch: 2016-07-22 13:58:01
connection accepted from ('140.138.142.18', 63686)
Predict Done
result = [0]
############################
Catch: 2016-07-22 14:03:01
connection accepted from ('140.138.142.18', 63751)
Predict Done
result = [0]
############################
Catch: 2016-07-22 14:08:01
connection accepted from ('140.138.142.18', 63789)
Predict Done
result = [0]
############################
```

Fig. 17. Server running output.

The prediction result is shown at Fig. 17. The script executes continuously as a system daemon that runs in background process, keeps tracking the hard drive condition.

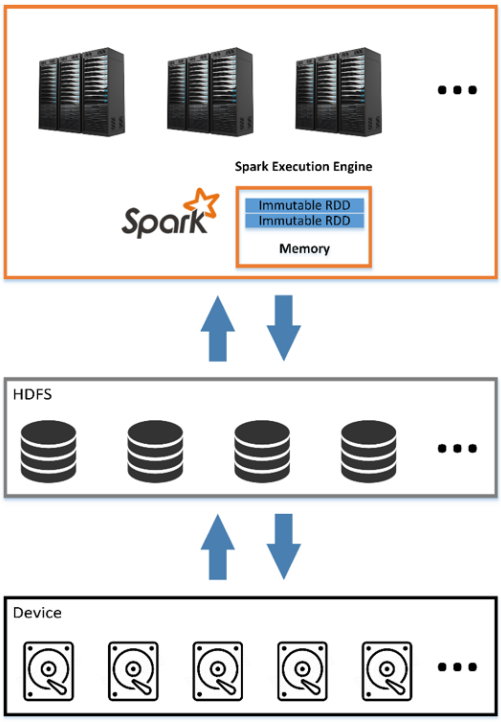

Fig. 18. The horizontal scalability of PAF-HD system

V. Conclusions

With the advent of cloud computing and online services, large enterprises rely on their data centers to serve end users. For the maintenance team, server and reliability are the first priority in order to provide stable and continuity of service. Among different servers, hard disk drives are known components to contribute significantly to server failures. However, after deducting vendor technical papers, there are only a few user experience studies which are essential in understanding device behavior in real-world deployments, probably due to the large number of devices needed to observe and the complex infrastructure required to track failures.

The development of the PAF-HD system will be a fundamental foundation when gathering device information and analyzing in an innovative way. It presents the random forest algorithm implement on Apache Spark for hard drive prediction. The system contains two stages: batch training and real-time predictions. Batch training section generates random forest models by using historical data. Real-time predictions stage uses the data collected from end-user device. A revised model trained every once in a whereas or on demand.

With the PAF-HD system, IT teams can quickly and easily connect, monitor devices and analyze data in real time. The machine learning solution in the PAF-HD system takes that data to provide a tool with a new intelligence that can improve efficiencies when maintaining a large scale storage system. Apache Spark delivers by persisting data in-memory to achieve up to 100 times faster queries while processing large datasets in Hadoop. This allows PAF-HD ideal for solving failure prediction problems in a large disk drive

population. Fig. 18 illustrates the horizontal scalability of the PAF-HD system.

The data we use in this thesis comes from real world data collection by Backblaze. It includes the SMART statistics which allows a disk drive to report data about its activity. We use machine learning to perform predictive monitoring and use the SMART attributes as input for training the machine learning model. The outcomes derived from the proposed framework of predictive monitoring for hard disk failure. The program runs as a daemon process and keeps tracking hard drive SMART statistics.

With the PAF-HD system, IT teams can quickly and easily connect, monitor devices and analyze data in real time. The machine learning solution in the PAF-HD system takes that data to provide IT staff with the ability to be more productive and proactive by helping them quickly identify the incoming failing.

VI. Limitations

In this research, we propose an innovative methodology for monitoring the hard drive failures. The PAF-HD system provides insights which are not provided by simple descriptive statistics or charts. Through the implementation process flow of this research, we use SMART statistics as evidence to determine if a drive is going to fail soon. However, the SMART stats are inconsistent from hard drive to hard drive. Every storage appliance vendor complies with common rules and defines a set of attributes, but the value meaning is entirely up to the drive manufacturer.

In other words, the PAF-HD system provides a hard drive dependent model. We cannot use a general predictive model to apply all hard drives. When analyzing different hard drives, we need to build new machine learning models. The training process requires historical data, experienced people turning and takes time to be optimized.

VII. Future Works

This research is the concept of Internet if Things (IoT) implementation, it takes the internet as the core, and applies to the physical device. In the coming years, the IOT will be fundamental in improving productivity and efficiency everywhere. A complete analyzing tool on the cloud which gathers data from multiple sources and provides support for multiple devices will lead to more management at any fields.

The main suggested future work is to incorporate time series methods into the training model. The SMART data is time series data, it is an ordered sequence of observations of a particular variable at evenly spaced time intervals. The failure pattern may not show in a single day, but time series extracts the subset of the SMART attributes observed between period of times. Once we discover the pre-failure pattern, we can use RF to perform pattern matching, check the given sequence of tokens for the presence of the constituents of the pre-failure pattern.

In this thesis, we only obtained hard drive data. To expand the reach of the proposed framework we suggest further research on a complete solution to connect plenty of components in data center, across any platform or operating system. Capture alarms and alerts from all of connected devices. Spot problem before they become an inconvenience. Take advantage of advanced analytics and machine learning to increase reliability and uptime. Decrease costly outages and expensive repairs with prescriptive maintenance.

Finally, connects to multiple components, which means there will be more different situations. Add more machine learning algorithms such as SVM or deep learning neural network. Provide more customized tool for user to select.

References

[1] J. Gu, N. Vichare, T. Tracy, and M. Pocht, "Prognostics implementation methods for electronics," presented at RAMS'07, Annual Reliability and Maintainability Symposium, 2007.

[2] V. Covic. *Backblaze Reveals Hard Drive Reliability Stats for Q3 2015*. (2015). [Online]. Available: http://www.cloudwards.net/news/backblaze-reveals-hard-drive-reliability-stats-for-q3-2015-10905/

[3] J.-C. Laprie, "Dependable computing and fault-tolerance," *Digest of Papers FTCS-15*, 1985, pp. 2-11.

[4] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, And think*, 2013, Houghton Mifflin Harcourt.

[5] N. M. Vichare and M. G. Pecht, "Prognostics and health management of electronics," *IEEE Transactions on Components and Packaging Technologies*, 2006, vol. 29, no. 1, pp. 222-229.

[6] H. Zhang *et al*., "Precursor parameter identification for power supply prognostics and health management," presented at 8th International Conference on Reliability, Maintainability and Safety, 2009.

[7] M. E. Edge *et al*., "Towards a proactive fraud management framework for financial data streams," presented at Third IEEE International Symposium on Dependable, Autonomic and Secure Computing, 2007.

[8] X. Lian *et al*., "Multiscale representations for fast pattern matching in stream time series," *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[9] A. H. R. Ko, A.-L. Jousselme, and P. Maupin, "A novel measure for data stream anomaly detection in a bio-surveillance systemm," in *Proc. the 14th International Conference on Information Fusion (FUSION)*, 2011, IEEE.

[10] S. Yang *et al*., "Detecting road traffic events by coupling multiple timeseries with a nonparametric bayesian method," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[11] G. Hamerly and C. Elkan, "Bayesian approaches to failure prediction for disk drives," *ICML*, 2001, Citeseer.

[12] G. F. Hughes *et al*., "Improved disk-drive failure warnings," *IEEE Transactions on Reliability*, 2002.

[13] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Hard drive failure prediction using non-parametric statistical methods," in *Proc. ICANN/ICONIP*, 2003, Citeseer.

[14] E. Pinheiro, W.-D. Weber, and L. A. Barroso, *Failure Trends in a Large Disk Drive Population*, 2007.

[15] V. Agarwal, C. Bhattacharyya, T. Niranjan, and S. Susarla, "Discovering rules from disk events for predicting hard drive failures," presented at International Conference on in Machine Learning and Applications, 2009, IEEE.

[16] W. L. Hosch. (2016). Machine learning. [Online]. Available: http://global.britannica.com/technology/machine-learning

[17] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, 1959, vol. 3, no. 3, pp. 210-229.

[18] T. M. Mitchell, *Machine Learning*, 1997, McGraw.

[19] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[20] L. Breiman and A. Cutler. Random forests leo breiman and adele cutler. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[21] L. Breiman, "Random forests," *Machine Learning*, 2001.

[22] T. K. Ho, "Random decision forests," in *Proc. the Third International Conference on Document Analysis and Recognition*, 1995, IEEE.

[23] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[24] L. Breiman, "Bagging predictors," *Machine Learning*, 1996.

[25] P. E. Utgoff, "Incremental induction of decision trees," *Machine Learning*, 1989.

[26] J. R. Quinlan, "Simplifying decision trees," *International Journal of Human-Computer Studies*, 1999.

[27] W. Yan, "Application of random forest to aircraft engine fault diagnosis," presented at IMACS Multiconference on in Computational Engineering in Systems Applications, 2006.
[28] S. Vincenzi *et al.*, "Application of a random forest algorithm to predict spatial distribution of the potential yield of Ruditapes philippinarum in the Venice lagoon, Italy," *Ecological Modelling*, 2011.
[29] Z. Li and Q. He, "Prediction of railcar remaining useful life by multiple data source fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2015.
[30] D. R. Gambill, W. A. Wall, A. J. Fulton, and H. R. Howard, "Predicting USCS soil classification from soil property variables using random forest," *Journal of Terramechanics*, 2016.
[31] A. Ng, *Machine Learning and AI via Brain simulations*, 2015, Stanford University.
[32] S. Miel, "What is the intuitive explanation of feature engineering in machine learning?" Quora, 2015.
[33] E Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," *FAST*, 2007.

Chuan-Jun Su is professor of the Industrial Engineering and Management Department at the Yuan Ze University, Taiwan. Prior to joining the University, he was an assistant professor with Hong Kong University of Science and Technology. His research interests include information systems, mobile agent technology, virtual reality, and intelligent design and manufacturing systems.

Jorge A. Quan Yon was born in Guatemala on August 8, 1984. Received his master's degree in industrial engineering and management at Yuan Ze University, Taiwan, in 2014. He is currently a Ph.D student at the Intelligent Information Systems laboratory at the Industrial Engineering and Management Department at Yuan Ze University, Taiwan. His research interests include Internet of things, big data analysis, social media prediction systems, and smart sustainable hydroponic systems.